

Отчёта по лабораторной работе №4

Создание и процесс обработки программ на языке ассемблера NASM

Барето Вилиан Мануел

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Программа Hello world!	6
3.2	Транслятор NASM	7
3.3	Расширенный синтаксис командной строки NASM	7
3.4	Компоновщик LD	8
3.5	Запуск исполняемого файла	8
3.6	Задание для самостоятельной работы	9
4	Выводы	11

Список иллюстраций

3.1	Создаем каталоги с помощью команды <code>mkdir</code>	6
3.2	Переходим в каталог с помощью команды <code>cd</code>	6
3.3	Создаем текстовый файл <code>hello.asm</code>	6
3.4	Открываем файл и заполняем его по примеру	7
3.5	Используем команду <code>nasm</code>	7
3.6	Проверяем работу команды	7
3.7	Преобразуем файл <code>hello.asm</code> в <code>obj.o</code>	7
3.8	Проверяем создание файла командой <code>ls</code>	7
3.9	Используем команду <code>ld</code>	8
3.10	Используем команду <code>ls</code>	8
3.11	Используем команду <code>ld</code> , создавая файл <code>main</code>	8
3.12	Используем команду <code>ls</code>	8
3.13	Используем команду <code>./hello</code>	8
3.14	Используем команду <code>cp</code>	9
3.15	Открываем файл в текстовом редакторе	9
3.16	Редактируем файл для своего имени и фамилии	9
3.17	Прописываем команды для работы файла и запускаем программу	9
3.18	Копируем файлы в каталог с ЛР4	9
3.19	Загружаем файлы	10

1 Цель работы

Освоить процедуры компиляции и сборки программ, познакомиться с языком ассемблера NASM.

2 Задание

Написать 2 программы(Hello world, lab4(Имя Фамилия))

3 Выполнение лабораторной работы

3.1 Программа Hello world!

Создаем каталог для работы с программами на языке ассемблера NASM (рис. fig. 3.1).

```
willianmanuelbarreto@fedora:~$ mkdir -p ~/work/arch-pc/lab04  
willianmanuelbarreto@fedora:~$
```

Рис. 3.1: Создаем каталоги с помощью команды mkdir

Переходим в созданный каталог (рис. fig. 3.2).

```
willianmanuelbarreto@fedora:~$ cd ~/work/arch-pc/lab04  
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$
```

Рис. 3.2: Переходим в каталог с помощью команды cd

Создаем текстовый файл (рис. fig. 3.3).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$ touch hello.asm  
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$
```

Рис. 3.3: Создаем текстовый файл hello.asm

Открываем данный файл в текстовом редакторе (рис. fig. 3.4).

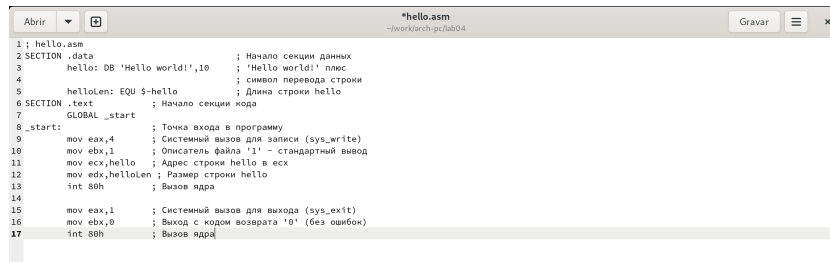


Рис. 3.4: Открываем файл и заполняем его по примеру

3.2 Транслятор NASM

Преобразуем текст программы в объектный код (рис. fig. 3.5).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$ nasm -f elf hello.asm
```

Рис. 3.5: Используем команду nasm

Проверяем создан ли объектный файл с помощью команды ls (рис. fig. 3.6).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$ ls
hello.asm hello.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$
```

Рис. 3.6: Проверяем работу команды

3.3 Расширенный синтаксис командной строки NASM

Компилируем исходный файл (рис. fig. 3.7).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$
```

Рис. 3.7: Преобразуем файл hello.asm в obj.o

Проверяем, как сработала команда (рис. fig. 3.8).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$ ls hello hello.asm hello.o list.lst obj.o
hello hello.asm hello.o list.lst obj.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$
```

Рис. 3.8: Проверяем создание файла командой ls

3.4 Компоновщик LD

Передаем объектный файл на обработку компоновщику (рис. fig. 3.9).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$
```

Рис. 3.9: Используем команду ld

Проверяем созданся ли исполняемый файл hello (рис. fig. 3.10).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$ ls hello hello.asm hello.o list.lst obj.o
hello hello.asm hello.o list.lst obj.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$
```

Рис. 3.10: Используем команду ls

Передаем объектный файл на обработку компоновщику (рис. fig. 3.11).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$
```

Рис. 3.11: Используем команду ld, создавая файл main

Проверяем созданся ли исполняемый файл hello (рис. fig. 3.12).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$ ls hello hello.asm hello.o list.lst main obj.o
hello hello.asm hello.o list.lst main obj.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$
```

Рис. 3.12: Используем команду ls

3.5 Запуск исполняемого файла

Запускаем на выполнение созданный исполняемый файл (рис. fig. 3.13).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$ ./hello Hello world!
Hello, world!
willianmanuelbarreto@fedora:~/work/arch-pc/lab04$
```

Рис. 3.13: Используем команду ./hello

3.6 Задание для самостоятельной работы

Создаем копию файла hello.asm (рис. fig. 3.14).

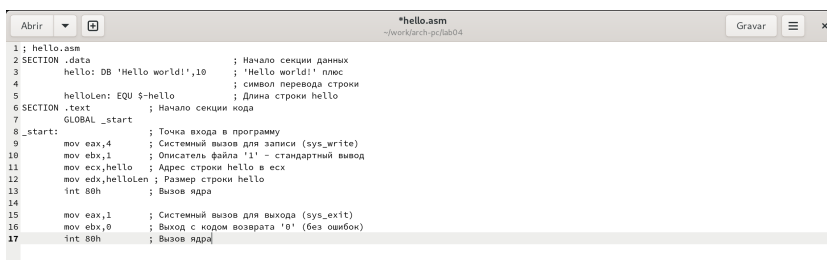
```
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$ cp hello.asm lab4.asm
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$
```

Рис. 3.14: Используем команду cp

Открываем файл и редактируем его (рис. fig. 3.15).

```
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$ gedit lab4.asm
```

Рис. 3.15: Открываем файл в текстовом редакторе



```
1: hello.asm
2: SECTION .data
3:     hello: DB 'Hello world!',10
4:     ; символ перевода строки
5:     helloLen: EQU $-hello
6: SECTION .text
7:     GLOBAL _start
8: _start:
9:     mov eax,4
10:    mov ebx,1
11:    mov ecx,hello
12:    mov edx,helloLen
13:    int 80h
14:    mov eax,1
15:    mov ebx,0
16:    int 80h
```

Рис. 3.16: Редактируем файл для своего имени и фамилии

Прописываем те же команды, что и с первой программой (рис. fig. 3.17).

```
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$ nasm -f elf lab4.asm
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o hello
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$ ./hello
Hello, world!
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$
```

Рис. 3.17: Прописываем команды для работы файла и запускаем программу

Копируем файлы в локальный репозиторий (рис. fig. 3.18).

```
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2024-2025/'Архитектура компьютера'/study_2024-2025_arh-pc/Labs/Lab04
willianmanuelbarreto@fedora: ~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2024-2025/'Архитектура компьютера'/study_2024-2025_arh-pc/Labs/Lab04
```

Рис. 3.18: Копируем файлы в каталог с ЛР4

Переходим в каталог лабораторных работ и загружаем файлы на Github (рис. fig. 3.19).

```

williammanuelbarreto@fedora: ~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04
cp: impossível criar o ficheiro normal: /home/williammanuelbarreto/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04: Ficheiro ou pasta inexistente
williammanuelbarreto@fedora: ~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab04
williammanuelbarreto@fedora: ~/work/arch-pc/lab04$ cp lab04.asm ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab04
williammanuelbarreto@fedora: ~/work/arch-pc/lab04$ cp ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab04
cp: operando de ficheiro destino em falta após '/home/williammanuelbarreto/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab04'
fente "cp --help" para mais informação.
williammanuelbarreto@fedora: ~/work/arch-pc/lab04$ cd ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc
williammanuelbarreto@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc$ git add .
williammanuelbarreto@fedora: ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-pc$ git commit -am 'feat(main): add files lab-4'
[master 88622e5] feat(main): add files lab-4
38 files changed, 43 insertions(+), 15 deletions(-)
create mode 100644 labs/lab01/report/report.docx
rename labs/lab02/report/image/placement_800_600_tech.jpg => lab01/report/report.pdf (91%)
create mode 100644 labs/lab03/report/image/10mb.png
create mode 100644 labs/lab03/report/image/11mb.png
create mode 100644 labs/lab03/report/image/12mb.png
create mode 100644 labs/lab03/report/image/13mb.png
create mode 100644 labs/lab03/report/image/14mb.png
create mode 100644 labs/lab03/report/image/15mb.png
create mode 100644 labs/lab03/report/image/16mb.png
create mode 100644 labs/lab03/report/image/17mb.png
create mode 100644 labs/lab03/report/image/18mb.png
create mode 100644 labs/lab03/report/image/19mb.png
create mode 100644 labs/lab03/report/image/20mb.png
create mode 100644 labs/lab03/report/image/21mb.png
create mode 100644 labs/lab03/report/image/22mb.png
create mode 100644 labs/lab03/report/image/23mb.png

```

Рис. 3.19: Загружаем файлы

4 Выводы

Мы познакомились с языком ассемблера NASM и создали две работающих программы.